HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200313534-2

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Dong WEI

Application No.: 10/737,111

Filing Date: December 15, 2003

Confirmation No.: 5499

Examiner: Elmira Mehrmanesh

Group Art Unit: 2113

Title: METHOD AND APPARATUS FOR PROVIDING UPDATED PROCESSOR POLLING

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on 8/15/2007.

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month $120	☐ 2nd Month $450	☐ 3rd Month $1020	☐ 4th Month $1590

☐ The extension fee has already been filed in this application.

☑ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of $500. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

☑ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to:
Commissioner for Patents, Alexandria, VA 22313-1450
Date of Deposit: 10/13/2007

OR

☐ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.
Date of facsimile:
Typed Name: Ilene L. Fish
Signature:

Respectfully submitted,
Dong WEI
By
John P. Wagner, Jr.
Attorney/Agent for Applicant(s)
Reg No. : 35,398
Date : 10/12/2007
Telephone : 408-377-0500

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Patent Application

Appellant:	Stephenson, Bryan	Confirmation No.:	5499
Application No.:	10/737,111	Group Art Unit:	2113
Filed:	December 14, 2003	Examiner:	Mehrmanesh, E.

For: Method and Apparatus for Providing Updated Processor Polling Information

APPEAL BRIEF

Table of Contents

I. Real Party in Interest

The assignee of the present application is Hewlett-Packard Development Company, L.P.

II. Related Appeals and Interferences

There are no related appeals or interferences known to the Appellant.

III. Status of Claims

Claims 1-20 are rejected. This Appeal involves Claims 1-20.

IV. Status of Amendments

All proposed amendments have been entered. An amendment subsequent to the Final Action has not been filed.

V. Summary of Claimed Subject Matter

Independent Claims 1, 12 and 23 of the present application pertains to a method and system for responding to network intrusions.

At least one embodiment of Claim 1 "A method for providing updated processor polling information" is depicted in Flowchart 4 and Figure 5. In one embodiment, as shown at least at 402 of Figure 4 and page 9 lines 11-18, processor polling information collected at boot time is provided to the operating system 510. Processor polling information may be collected by each processor 104 within the integrated processor platform 100, for presentation to the operating system 510. The operating system 510 obtains the collected processor polling data and stores it in its own database for future use. At 404 of Figure 4 and page 10 lines 24-29 the system remains in a loop, until the occurrence of an event that may have potentially altered the state of the system. An example of such an event may be the addition, deletion or deconfiguration of processors 104 during runtime. At 406 of Figure 4 and page 11 lines 4-8, upon occurrence of an event with the potential of altering the state of the system, the operating system 510 is notified. At 408 of Figure 4 and page 11 lines 9-15, one embodiment provides updated processor polling information during runtime to said operating system 510, said updated processor polling information reflecting operating conditions of said integrated processor system after the occurrence of the triggering event.

In Claim 12 and at Figure 5, "a computer program embodied on a computer readable medium for providing updated processors polling information" is recited and depicted in Flowchart 5. In one embodiment, as shown at least at 510 of Figure 5 and page 12 lines 22 through page 13 line 7, in one embodiment the computer program causing a computer to create a processor polling information table, said processor polling information table (such as shown in Figures 2 and 3 and described on page 7 lines 12-23) being populated with boot time processor polling information, wherein said processor polling information describes operating conditions of an integrated processing system 100 of Figure 1. For example, in one embodiment of the present invention, at boot time, the CPEP table creator 502 collects system data to be organized into a CPEP table structure, for presentation and use by the operating system 510. The CPEP table data is provided to the operating system 510 during boot time, at firmware handoff to the operating system 510.

In addition, the computer program also causes the computer to update said processor polling information table upon receipt of a notification that a triggering event has occurred, wherein said triggering event may potentially alter said operating conditions of said integrated processor system. For example, at 510 of Figure 5 and page 12 lines 22 through page 13 line 7 the triggering event detector 506 detects the occurrence of an event that may potentially affect the platform configuration and polling for CPEs. Upon the occurrence of the detected event(s), the triggering event detector 506 notifies the operating system 510 through a CPEP table updator 504 that the boot time information received through the CPEP table creator 502 may have become stale.

In Claim 16, "an apparatus for updating processor polling information" is recited and depicted in Flowchart 5. For example, at 510 of Figure 5 and page 12 lines 22 through page 13 line 7, a corrected platform error polling (CPEP) table creator 502 for creating a CPEP table coupled to an operating system 510, said CPEP table (as shown in Figures 2 and 3) being populated with boot time processor polling information, wherein said processor polling information describes operating conditions of an integrated processor system. In one embodiment, such as shown in Figure 5 and page 13 lines 3-8, a triggering event detector 506 is coupled to said operating system 510, said triggering event detector 506 capable of detecting an occurrence of a triggering event, where said triggering event may potentially alter said operating conditions of said integrated processor system.

Additionally, such as shown in Figure 5 and page 13 lines 10-20, in one embodiment a CPEP table updator 504 is also coupled to said operating system 510 and further coupled to said triggering event detector 506, wherein, upon a receipt of a notification of an occurrence of a triggering event from said triggering event detector 506, said CPEP table updator 504 provides updated processor polling information to said operating system 510 based on said altered operating conditions.

VI. Grounds of Rejection to Be Reviewed on Appeal

1. Claims 1 and 3-8 stand rejected under 35 U.S.C. §102(b) as being anticipated by Takashima et al. (6,347,372) referred to hereinafter as "Takashima".

2. Claims 9-20 stand rejected under 35 U.S.C. §102(e) as being anticipated by Schultz et al. (6,948,094) referred to hereinafter as "Schultz".

3. Claim 2 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Takashima in view of Schultz.

VII. Argument

1. Whether Claims 1 and 3-8 are anticipated by Takashima under 35 U.S.C. § 102(b).

A. Claim Features are not Met by the Cited References

Appellant respectfully submits that the rejection of Claims 1 and 3-8 is improper as the rejection of Claims 1 and 3-8 does not satisfy the requirements of a *prima facie* case of anticipation under 35 U.S.C. § 102(b) as claim features are not met by the cited reference.

Appellant respectfully submits that Claim 1 recites,

> A method of providing updated processor polling information, comprising:
> collecting processor polling information at boot time to be provided to an operating system, said processor polling information describing operating conditions of an integrated processing system;
> notifying said operating system that a triggering event has occurred, wherein said triggering event potentially alters said operating conditions of said integrated processor system; and
> providing updated processor polling information during runtime to said operating system, said updated processor polling information reflecting operating conditions of said integrated processor system after the occurrence of the triggering event.

Appellant respectfully submits that Takashima does not teach or suggest any of the limitations "collecting processor polling information at boot time to be provided to an operating system, said processor polling information describing operating conditions of an integrated processing system; notifying said operating system that a triggering event has occurred, wherein said triggering event potentially alters said operating conditions of said integrated processor system; and providing updated processor polling information during runtime to said operating system, said updated processor polling information reflecting operating conditions of said integrated processor system after the occurrence of the triggering event," (emphasis added).

Referring to the last sentence of the abstract, Takashima teaches loading boot data into a boot processor via a shared bus. For example, the abstract states from line 6 to the last line of the abstract,

> The processors constitute at least one boot processor to which the boot data is to be loaded. The boot control device includes a time slot division unit which produces time slots on the shared bus by multiplexing channels for the processors, and a time sharing control unit which determines a time slot for the boot processor among the time slots produced, and assigns the time slot to the boot processor. The time sharing control unit includes a processor interface part which notifies a time-slot location of the time slot determined, to the boot processor. A bus time-slot setting part notifies the time-slot location to the storage device, allowing reading of the boot data from the storage device and inserting of the boot data into the time slot at the time-slot location among the time slots on the shared bus, so that the boot data is loaded into the boot processor via the shared bus (emphasis added).

The Office Action asserts that Takashima teaches "collecting processor polling information at boot time to be provided to an operating system, said processor polling information describing operating conditions of an integrated processing system," (emphasis added) at element 36 on FIG. 4. Referring to FIG. 4, element 36 is a polling control unit (POLC) which is a part of the boot control device 45. Col. 6 lines 34-39 of Takashima clearly indicates that the boot control device 45 is a piece of hardware. There is nothing in Figure 4 or anywhere in Takashima that indicates that Takashima teaches "collecting processor polling information at boot time to be provided to an operating system." Therefore Takashima does not teach or suggest "collecting processor polling information at boot time to be provided to an operating system, said processor polling information describing operating conditions of an integrated processing system," (emphasis added) as recited by Claim 1.

Furthermore, the Office Action asserted that Takashima teaches "notifying said operating system that a triggering event has occurred, wherein said triggering event potentially alters said operating conditions of said integrated processor system," at Col. 8 lines 2-9. Col. 8 lines 2-9 state,

> When an error occurs in one of the "N" processors of the processor block 31, the CC 41 notifies the TSC 33 that the TSC 33 should avoid the assignment of the time slot to the defective processor. The CC 41 causes the TSC 33 to notify a host system, which manages the processor of concern, that the channel for the processor of concern is defective, or that the channel for the processor of concern should be set in an OFF status. (emphasis added).

It is not clear to Appellant what in Col. 8 lines 2-9 the Office Action asserts teaches an "operating system." The CC 41 and the TSC 33 are a part of the boot control device 45 which as already established herein is a piece of hardware. Takashima does not clarify what

his "host system" is. Referring to the abstract, Takashima teaches loading boot data into a boot processor via the shared bus. Therefore, Takashima's host system is clearly not the operating system. Further, Col. 9 lines 12-13 state, "Next, the TSC 33 notifies a boot start timing...," which occurs after the processing described Col. 8 lines 2-9. Col. 8 lines 2-9, which refers to notifying a "host system." Part of the boot process **is** to boot the operating system. Therefore, the operating system **cannot** be notified of anything until the boot process has completed. Since Col. 9 lines 12-13 indicate that the boot process has not completed yet, the "***host system***" cannot be an operating system or a part of an operating system because the boot process hasn't been completed (emphasis added).

The Office Action asserts that Takashima teaches "providing updated processor polling information during runtime to said operating system, said updated processor polling information reflecting operating conditions of said integrated processor system after the occurrence of the triggering event," (emphasis added) at Col. 7 lines 43-47 and Col. 7 lines 26-42. Col. 7 lines 43-46 states, "The external port interface unit (EX/IF) 38 sends an external signal to the storage device 42 in order to update the information stored in the storage device 42" (emphasis added). Note that Col. 7 lines 43-46 say nothing about an operating system or during run time. Clearly the storage device 42 is not an operating system. Further Col. 7 lines 26-42 say nothing about an operating system or during run time, among other things. Col. 7 lines 26-42 refer to a POLC 36, a TSC 33 and a P/SD 35, which are all part of the boot control device 45. As already established the boot control device 45 is hardware not an operating system.

For the foregoing reasons, Claim 1 should be patentable. Claims 2-8 depend on Claim 1. These dependent claims include all of the limitations of their respective independent claims. Further, these dependent claims include additional limitations which further make them patentable. Therefore, these dependent claims should be patentable for at least the reasons that their respective independent claims should be patentable.

2. Whether Claims 9-20 are anticipated by Shultz under 35 U.S.C. § 102(e).

A. Claim Features are not Met by the Cited References

Appellant respectfully submits that the rejection of Claims 9-20 is improper as the rejection of Claims 9-20 does not satisfy the requirements of a *prima facie* case of anticipation under 35 U.S.C. § 102(e) as claim features are not met by the cited reference.

Claim 9 (and similarly Claim 16) recites,

> A computer program embodied on a computer readable medium for providing updated processors polling information, the computer program causing a computer to perform the steps of:
>
> creating a processor polling information table, said processor polling information table being populated with boot time processor polling information, wherein said processor polling information describes operating conditions of an integrated processing system; and
>
> updating said processor polling information table upon receipt of a notification that a triggering event has occurred, wherein said triggering event may potentially alter said operating conditions of said integrated processor system (emphasis added).

Appellant respectfully submits that Schultz does not teach or suggest any of the limitations of Claim 9.

Schultz teaches a method of correcting a machine check error based on the interrupt model. For example, referring to Col. 3 lines 32-46, Schultz states,

> ...When the platform or processor hardware 101 generates a machine check 200, control is passed to a processor error handler 204 in the PAL 104. In turn, control is passed to the platform error handler 206 in the SAL 106. Control may in turn be passed to the OS machine check handler 206 in the SAL 106. Control may in turn be passed to the OS machine check handler 10 in the operating system software 110. If the error is corrected, control will be returned to the interrupted processor context. Otherwise, the system will be halted or rebooted.
>
> The machine check architecture error handling model consists of different software components that work in close cooperation to handle different error conditions. PAL, SAL, and the operating system have error handling components, which are tightly coupled through a well defined interface.

Machine checks are a type of interrupt. Col. 3 lines 32-46 clearly state that PAL, SAL and the operating system are tightly coupled through the use of an interrupt type model.

In contrast, Claim 9 recites "polling," which is fundamentally different from an interrupt.

Further, the Office Action points out that Schultz teaches "[T]he operating system may disable this automatic interrupt notification and periodically poll the firmware to collect corrected error events."

However, in contrast, Appellant respectfully points out that Claim 9 clearly recites "creating a processor polling information table, said processor polling information table being populated with boot time processor polling information."

This is very different from "periodically polling the firmware" as anticipated by Shultz. In fact, the deficiencies of Shultz teachings are clearly provided in the background of the Instant Application. Specifically, the instant application explains this fundamental difference starting in the last paragraph on page 5 through the second paragraph on page 6. In referring to this portion of the instant application, Appellant is not attempting to read portions of the instant application into the claims. Claim 9 clearly recites "polling." Appellant is merely pointing out the portion of the Specification within which the fundamental difference between polling, which Claim 9 recites, and interrupts, which Schultz teaches is provided.

In other words, Claim 9 provides a distinctly different approach for monitoring the health of a system by creating the processor polling information table to allow the system to not have to periodically poll the entire firmware.

For these reasons, Appellant respectfully submits that Claims 9 and 16 should be patentable. Claim 10-15 depend on Claim 9. Claims 17-20 depend on Claim 16. These dependent claims include all of the limitations of their respective independent claims. Further, these dependent claims include additional limitations which further make them patentable. Therefore, these dependent claims should be patentable for at least the reasons that their respective independent claims should be patentable.

3. Claim 2 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Takashima in view of Schultz.

A. Claim Features are not Met by the Cited References

Appellant respectfully submits that the rejection of Claim 2 is improper as the rejection of Claim 2 does not satisfy the requirements of a *prima facie* case of obviousness under 35 U.S.C. § 103(a) as claimed features are not met by the cited reference.

For the reasons provided herein, Appellant respectfully submits that Claim 1 is patentable. Claim 2 depends on Claim 1. As such, Claim 2 includes all of the limitations of its respective independent claim. Further, Claim 2 includes additional features which further make it patentable. Therefore, Claim 2 should be patentable for at least the reasons that the respective independent Claim 1 should be patentable.

Response to Arguments

In the first paragraph of the response to arguments section, on pages 10, there is the statement that Takashima teaches "notifying said operating system that a triggering event has occurred, wherein said triggering event potentially alters said operating conditions of said integrated processor system" at col. 9 lines 56-67 through col. 10 lines 1-12. However, Appellant respectfully disagrees.

Specifically, as the Office Action states, Takashima teaches "in the event of error detection, the defective processor is identified and isolated by setting its channel to an OFF status (col. 10, lines 4-25).

In contrast, Claim 1 clearly states "notifying said operating system that a triggering event has occurred, wherein said triggering event potentially alters said operating conditions of said integrated processor system" (emphasis added).

Moreover, Claim 1 goes on to state "providing updated processor polling information during runtime to said operating system, said updated processor polling information reflecting

operating conditions of said integrated processor system after the occurrence of the triggering event" (emphasis added).

However, Takashima would not anticipate the claimed feature because there would be no need to provide updated information reflecting the operating conditions. Instead, as Takashima is clearly stated by the present Office Action, "in the event of error detection, the defective processor is identified and isolated by setting its channel to an **OFF** status" (emphasis added).

Thus, there would be no need for providing updated processor polling information during runtime to the operating system since the operating system would already know, ***without requiring any polling***, the defective processor was OFF.

Therefore, Appellant points out that Takashima does not anticipate the features as recited by Claim 1. As such, Appellant respectfully submits Claim 1 overcomes the rejection under 35 U.S.C. § 102(b) and requests that the rejection of Claim 1 be overturned.

In the third paragraph of the response to arguments section, on pages 10, there is the statement that Schultz teaches "[T]he operating system may disable this automatic interrupt notification and periodically poll the firmware to collect corrected error events."

In contrast, Appellant respectfully points out that Claim 9 includes the features "creating a processor polling information table, said processor polling information table being populated with boot time processor polling information, wherein said processor polling information describes operating conditions of an integrated processing system; and updating said processor polling information table upon receipt of a notification that a triggering event has occurred, wherein said triggering event may potentially alter said operating conditions of said integrated processor system."

This is very different from "periodically polling the firmware" as anticipated by Shultz. In fact, the deficiencies of Shultz teachings are clearly provided in the background of the Instant Application. Specifically, the reason for the creating of the processor polling information table is to allow the system to not have to periodically poll the entire firmware.

Therefore, Appellant points out that Schultz does not anticipate the features as recited by Claims 9 and 16. As such, Appellant respectfully submits Claims 9 and 16 overcome the rejection under 35 U.S.C. § 102(e) and request that the rejection of Claims 9 and 16 be overturned.

In summary, Appellant respectfully submits that the Examiner's rejections of the Claims are improper as the rejection of Claims 1-20 does not satisfy the requirements of a *prima facie* case of anticipation or obviousness as claim features are not met by the cited references. Accordingly, Appellant points out that the rejections of Claims 1-20 under 35 U.S.C. §102(b), 35 U.S.C. §102(e) and 35 U.S.C. §103(a) are improper and should be reversed.

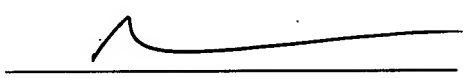
Conclusion

Appellant believes that pending Claims 1-20 are directed toward patentable subject matter. As such, Appellant submits that Claims 1-20 are patentable and respectfully request that the rejection of Claims 1-20 be reversed.

The Appellant wishes to encourage the Examiner or a member of the Board of Patent Appeals to telephone the Appellant's undersigned representative if it is felt that a telephone conference could expedite prosecution.

Respectfully submitted,
Wagner Blecher LLP

Dated: 10/12/07



John P. Wagner, Jr.
Registration No.: 35,398

Wagner Blecher LLP
Westridge Business Park
123 Westridge Drive
Watsonville, CA 95076

Phone: (408) 377-0500

VIII. Appendix - Clean Copy of Claims on Appeal

1. (original) A method for providing updated processor polling information comprising:

collecting processor polling information at boot time to be provided to an operating system, said processor polling information describing operating conditions of an integrated processing system;

notifying the operating system that a triggering event has occurred, wherein said triggering event potentially alters said operating conditions of said integrated processor system; and

providing updated processor polling information during runtime to said operating system, said updated processor polling information reflecting operating conditions of said integrated processor system after the occurrence of the triggering event.

2. (original) The method of claim 1 further comprising:

creating a corrected platform error polling (CPEP) table, wherein said CPEP table is populated with processor polling information collected at boot time.

3. (original) The method of claim 1 wherein the triggering event is based on an addition of a processor device.

4. (original) The method of claim 1 wherein the triggering event is based on a deletion of a processor device.

5. (original) The method of claim 1 wherein the triggering event is based on a deconfiguration of a processor device.

6. (original) The method of claim 1 further comprising:

performing a process on an object associated with a processor device and returning a value to an operating system of said integrated processor system, wherein said value supercedes a corresponding processor polling information.

7. (original) The method of claim 6 wherein the value that is returned is a zero indicating that the corresponding processor device is not to be polled.

8. (original) The method of claim 7 wherein the value that is returned is a non-zero number indicating a minimum polling frequency

9. (original) A computer program embodied on a computer readable medium for providing updated processors polling information, the computer program causing a computer to perform the steps of:

creating a processor polling information table, said processor polling information table being populated with boot time processor polling information, wherein said processor polling information describes operating conditions of an integrated processing system; and

updating said processor polling information table upon receipt of a notification that a triggering event has occurred, wherein said triggering event may potentially alter said operating conditions of said integrated processor system.

10. (original) The computer program of claim 9 wherein said computer program further causes said computer to:

invoke a bus check notification upon an online addition of a processor device, wherein said bus check notification indicates to an operating system that a re-enumeration of a device tree needs to be performed, and wherein said operating system invokes a Poll for corrected Platform Error (_PPE) procedure that returns a value indicating a polling frequency for said added processor device.

11. (original) The computer program of claim 9 wherein said computer program further causes a computer to:

invoke an eject request notification upon an online deletion of a processor device, wherein said eject request notification indicates to an operating system to update its CPEP table and not poll from said processor device which has been deleted.

12. (original) The computer program of claim 9 wherein said computer program further causes a computer to:

invoke a CPEP check notification invoked by an online deconfiguration of a faulty processor device, wherein the CPEP check notification indicates to an operating system to invoke a _PPE procedure indicating to said operating system alternative processor devices to be polled.

13. (original) The computer program of claim 9 wherein said computer program further causes a computer to:

invoke a _PPE procedure object associated with a processor device, wherein said _PPE procedure object returns a value that supercedes a corresponding CPEP table processor polling information.

14. (original) The computer program of claim 13 wherein a zero return value indicates that said corresponding processor is not to be polled.

15. (original) The computer program of claim 13 wherein a non-zero return value indicates a minimum polling frequency.

16. An apparatus for updating processor polling information comprising:

a corrected platform error polling (CPEP) table creator for creating a CPEP table coupled to an operating system, said CPEP table being populated with boot time processor polling information, wherein said processor polling information describes operating conditions of an integrated processor system;

a triggering event detector coupled to said operating system, said triggering event detector capable of detecting an occurrence of a triggering event, where said triggering event may potentially alter said operating conditions of said integrated processor system; and

a CPEP table updator coupled to said operating system and further coupled to said triggering event detector, wherein, upon a receipt of a notification of an occurrence of a triggering event from said triggering event detector, said CPEP table updator provides updated processor polling information to said operating system based on said altered operating conditions.

17. (original) The apparatus of claim 16 wherein the triggering event detector is configured to detect an event triggered by an addition or deletion of a processor device.

18. (original) The apparatus of claim 16 wherein the triggering event detector is further configured to detect an event based on a deconfiguration of a processor device.

19. (original) The apparatus of claim 16 further comprising:

a polling frequency calculator coupled to said CPEP table updator, said polling frequency calculator configured to return a value that indicates a minimum polling frequency for a selected processor device.

20. (original) The apparatus of claim 19 wherein said polling frequency calculator is configured to forgo polling said selected processor device when said polling frequency calculator returns a zero value for said selected processor device.

IX. Evidence Appendix

No evidence is herein appended.

X. Related Proceedings Appendix

No related proceedings.